Recursion

Recursion is a technique where a function calls itself to solve a problem by breaking it down into smaller, similar subproblems.

Key Concepts

- Base case: The condition that stops the recursion
- Recursive case: The part where the function calls itself
- Call stack: How the computer keeps track of function calls

Recursion vs Iteration

Recursion can often be replaced with iteration (loops), but recursive solutions can be more elegant for problems that have a natural recursive structure.

Example

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
```

This function calculates the factorial of a number by calling itself with a smaller value until it reaches the base case.

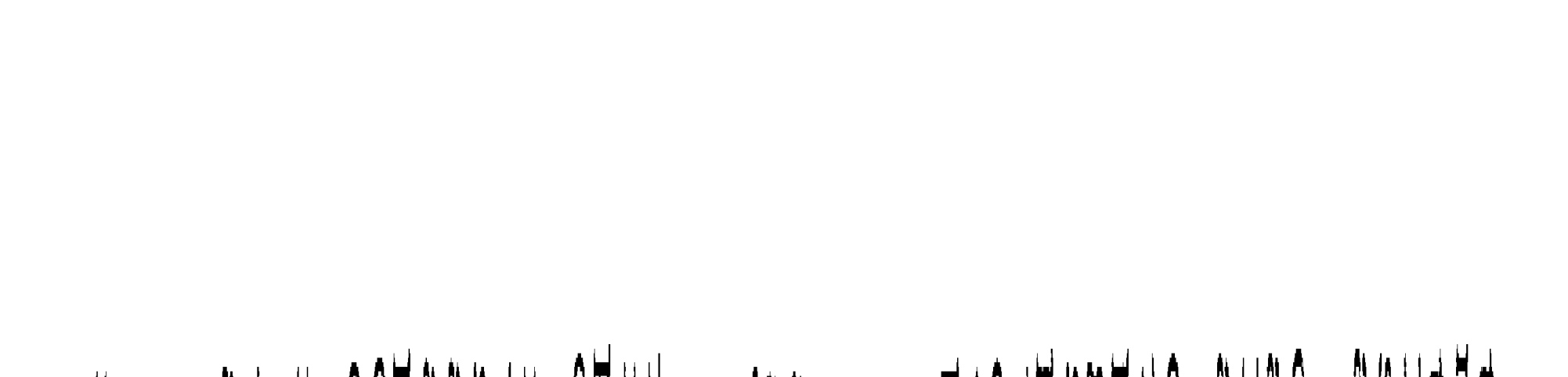

Table of Contents

Introduction 1

Chapter 1: Getting Started 5

Chapter 2: Basic Concepts 12

Chapter 3: Advanced Techniques 28

Chapter 4: Troubleshooting 45

Appendix A: Glossary 60

Appendix B: Resources 72

Index 80

Contents

Introduction 1

Chapter 1 Introduction to Computer Science 3

Chapter 2 Programming Languages 15

Chapter 3 Data Structures and Algorithms 27